Expert Systems with Applications 38 (2011) 11630–11640

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Interestingness measures for association rules: Combination between lattice and hash tables[☆]

Bay Vo[a,*], Bac Le[b]

[a] Department of Computer Science, Ho Chi Minh City University of Technology, Ho Chi Minh, Viet Nam
[b] Department of Computer Science, University of Science, Ho Chi Minh, Viet Nam

ARTICLE INFO

Keywords:
Association rules
Frequent itemsets
Frequent itemsets lattice
Hash tables
Interestingness association rules
Interestingness measures

ABSTRACT

There are many methods which have been developed for improving the time of mining frequent itemsets. However, the time for generating association rules were not put in deep research. In reality, if a database contains many frequent itemsets (from thousands up to millions), the time for generating association rules is more longer than the time for mining frequent itemsets. In this paper, we present a combination between lattice and hash tables for mining association rules with different interestingness measures. Our method includes two phases: (1) building frequent itemsets lattice and (2) generating interestingness association rules by combining between lattice and hash tables. To compute the measure value of a rule fast, we use the lattice to get the support of the left hand side and use hash tables to get the support of the right hand side. Experimental results show that the mining time of our method is more effective than the method that of directly mining from frequent itemsets uses hash tables only.

1. Introduction

Since the mining association rules problem presented in 1993 (Agrawal, Imielinski, & Swami, 1993), there have been many algorithms developed for improving the effect of mining association rules such as Apriori (Agrawal & Srikant, 1994), FP-tree (Grahne & Zhu, 2005; Han & Kamber, 2006; Wang, Han, & Pei, 2003), and IT-tree (Zaki & Hsiao, 2005). Although the approaches for mining association rules are different, their processing ways are nearly the same. Their mining processes are usually divided into the following two phases:

(i) Mining frequent itemsets;
(ii) Generating association rules from them.

Recent years, some researchers have studied about interestingness measures for mining interestingness association rules (Aljandal, Hsu, Bahirwani, Caragea, & Weninger, 2008; Athreya & Lahiri, 2006, Bayardo & Agrawal, 1999; Brin, Motwani, Ullman, & Tsur, 1997; Freitas, 1999; Holena, 2009; Hilderman & Hamilton, 2001; Huebner, 2009; Huynh et al., 2007, chap. 2; Lee, Kim, Cai, & Han, 2003; Lenca, Meyer, Vaillant, & Lallich, 2008; MCGarry, 2005; Omiecinski, 2003; Piatetsky-Shapiro, 1991; Shekar & Natarajan, 2004; Steinbach, Tan, Xiong, & Kumar, 2007; Tan, Kumar, & Srivastava, 2002; Waleed, 2009; Yafi, Alam, & Biswas, 2007; Yao, Chen, & Yang, 2006). A lot of measures have been proposed such as support, confidence, cosine, lift, chi-square, gini-index, Laplace, phi-coefficient (about 35 measures Huynh et al., 2007). Although they differ from the equations, they use four elements to compute the measure value of rule $X \rightarrow Y$: (i) n; (ii) n_X; (iii) n_Y; and (iv) n_{XY}, where n is the number of transactions, n_X is the number of transactions containing X, n_Y is the number of transactions containing Y, n_{XY} is the number of transactions containing both X and Y. Some other elements for computing the measure value are determined via n, n_X, n_Y, n_{XY} as follows: $n_{\overline{X}} = n - n_X$, $n_{\overline{Y}} = n - n_Y$, $n_{X\overline{Y}} = n_X - n_{XY}$, $n_{\overline{X}Y} = n_Y - n_{XY}$, and $n_{\overline{XY}} = n - n_{XY}$.

We have n_X = support (X), n_Y = support (Y), and n_{XY} = support (XY). Therefore, if support (X), support (Y), and support (XY) are determined then value of all measures of a rule will be determined.

We can see that almost previous studies were done in small databases. However, databases are often very large in practice. For example, Huynh et al. only mined in the databases which numbers of rules are small (contain about one hundred thousand rules, Huynh et al., 2007). In fact, there are a lot of databases containing about millions of transactions and thousands items containing millions of rules, the time for generating association rules and computing their measure values is very long. Therefore, this paper proposes a method for computing the interestingness measure

[☆] This work was supported by Vietnam's National Foundation for Science and Technology Development (NAFOSTED), project ID: 102.01-2010.02.

[*] Corresponding author. Tel.: +84 08 39744186.
E-mail addresses: vdbay@hcmhutech.edu.vn (B. Vo), lhbac@fit.hcmus.edu.vn (B. Le).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.03.042

Table 1
An example database.

TID	Item bought
1	A, C, T, W
2	C, D, W
3	A, C, T, W
4	A, C, D, W
5	A, C, D, T, W
6	C, D, T

Table 2
Value of some measures with rule $X \to Y$.

Measures	Equations	Values
Confidence	$\frac{n_{XY}}{n_X}$	$\frac{3}{4}$
Cosine	$\frac{n_{XY}}{\sqrt{n_X n_Y}}$	$\frac{3}{\sqrt{4*3}} = \frac{3}{\sqrt{12}}$
Lift	$\frac{n_{XY} n}{n_X n_Y}$	$\frac{3*6}{4*3} = \frac{3}{2}$
Rule interest	$n_{XY} - \frac{n_X n_Y}{n}$	$3 - \frac{4*3}{6} = 1$
Laplace	$\frac{n_{XY}+1}{n_X+2}$	$\frac{4}{6}$
Jaccard	$\frac{n_{XY}}{n_X+n_Y-n_{XY}}$	$\frac{3}{4+3-3} = \frac{3}{4}$
Phi-coefficient	$\frac{n_{XY} n - n_X n_Y}{\sqrt{n_X n_Y n_{\overline{X}} n_{\overline{Y}}}}$	$\frac{3*6-4*3}{\sqrt{4*3*2*3}} = \frac{6}{\sqrt{72}}$

Table 3
Frequent itemsets from Table 1 with minSup = 50%.

FIs	Support
A	4
C	6
D	4
T	4
W	5
AC	4
AT	3
AW	4
CD	4
CT	4
CW	5
DW	3
TW	3
ACT	3
ACW	4
ATW	3
CDW	3
CTW	3
ACTW	3

```
Input: Database D and the minimum support threshold minSup
Output: FIL that contains all frequent itemsets of D
Method:
LATTICE_FIs ( )
    $l_r = \varnothing$
    $[\varnothing] = \{ l_i \times t(l_i) \mid l_i \in I \wedge \sigma(l_i) \ge \text{minSup} \}$
    for all $l_i \in [\varnothing]$ do
        $l_r$.AddChildren ({$l_i$})
    ENUMERATE_LATTICE([$\varnothing$])
    return $l_r$

ENUMERATE_LATTICE([$P$])
    for all $l_i \in [P]$ do
        $[P_i] = \varnothing$
        for all $l_j \in [P]$, with j > i do
            $I = l_j$
            $T = t(l_i) \cap t(l_j)$
            if $|T| \ge$ minSup then
                $[P_i] = [P_i] \cup \{ I \times T \}$
                UPDATE_LATTICE({$l_i$}, {$I$})
                {$l_i$}.AddChildren ({$I$})
                {$l_j$}.AddChildren ({$I$})
        ENUMERATE_LATTICE([$P_i$])
```

Fig. 1. An algorithm for building frequent itemsets lattice (Vo & Le, 2009).

Table 4
Hash tables for frequent itemsets in Table 3.

1	Value	A	C	D	T	W			
	Key	1	2	3	4	5			
2	Value	AC	AT	AW	CD	CT	CW	DW	TW
	Key	3	5	6	5	6	7	8	9
3	Value	ACT	ACW	ATW	CDW	CTW			
	Key	7	8	10	10	11			
4	Value	ACTW							
	Key	12							

Table 5
Hash tables for frequent itemsets in Table 3 when we use prime numbers as the keys.

1	Value	A	C	D	T	W			
	Key	2	3	5	7	11			
2	Value	AC	AT	AW	CD	CT	CW	DW	TW
	Key	5	9	13	8	10	14	16	18
3	Value	ACT	ACW	ATW	CDW	CTW			
	Key	12	16	20	19	21			
4	Value	ACTW							
	Key	33							

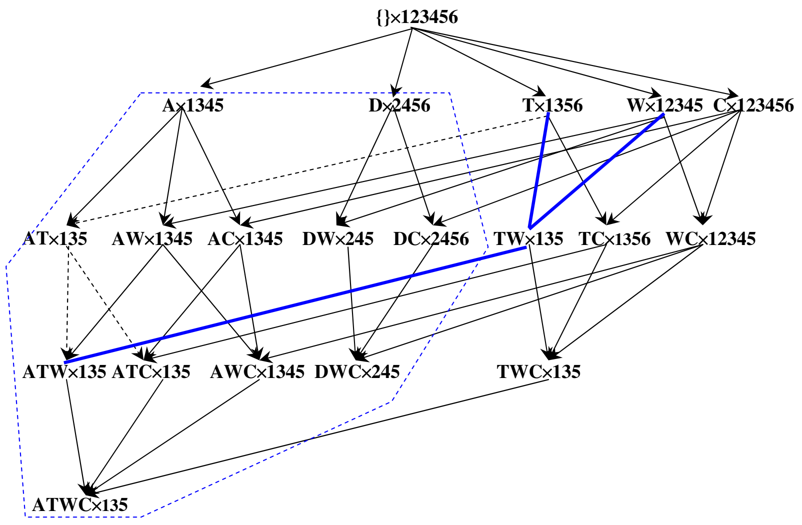


Fig. 2. Results of producing frequent itemset lattice from database in Table 1 with minSup = 50% ((Vo & Le, 2009).

```
Input: Frequent itemsets lattice ($L$, $\le$) with the root node $L_r$
Output: Association rules and their measure values
Method:

AR_LATTICE( )
  ARs = ∅
  for each $L_c \in L_r$.Children do
       EXTEND_AR_LATTICE( $L_c$ )
  return ARs

EXTEND_AR_LATTICE( $L_c$ )
  if $L_c$.flag = False then
       ENUMERATE_AR($L_c$)
       $L_c$.flag = True
       for each $L_s \in L_c$.Children do
            EXTEND_AR_LATTICE( $L_s$ )

ENUMERATE_AR($L_c$)
    Queue = ∅
    $n_X$ = support($L_c$)
    for all $L_s \in L_c$.Children do
         Add $L_s$ to Queue
         Mark $L_s$
  while Queue ≠ ∅ do
      $L$ = Get an element from Queue
      $n_{XY}$ = support($L$)
      $n_Y$ = Get the support from the hash table $|L \setminus L_c|^{th}$ with key = $\sum_{y \in L \setminus L_c} y$
      ARs = ARs ∪ { $L_c \to L \setminus L_c$ (supp($L$), $vm(n, n_X, n_Y, n_{XY})$}
      for all $L_s \in L$.Children do
            if $L_s$ is not marked then
                Add $L_s$ to Queue
                Mark $L_s$
```

Fig. 3. Generating association rules with interestingness measures using lattice and hash tables.

Table 7
Features of experimental databases.

Database	#Trans	#Items
Mushroom	8124	120
Chess	3196	76
Pumsb*	49046	7117
Retail	88162	16469
Accidents	340183	468

Table 8
Numbers of frequent itemsets and numbers of rules in databases correspond to their minimum supports.

Databases	minSup (%)	#FIs	#rules
Mushroom	35	1189	21522
	30	2735	94894
	25	5545	282672
	20	53583	19191656
Chess	80	8227	552564
	75	20993	2336556
	70	48731	8111370
	65	111239	26238988
Pumsb*	50	679	12840
	45	1913	53614
	40	27354	5659536
	35	116747	49886970
Retail	0.7	315	652
	0.5	580	1382
	0.3	1393	3416
	0.1	7586	23708
Accidents	50	8057	375774
	45	16123	1006566
	40	32528	2764708
	35	68222	8218214

values of association rules fast. We use lattice to determine itemsets *X*, *XY* and their supports. To determine the support of *Y*, we use hash tables.

The rest of this paper is as follows: Section 2 presents related works of interestingness measures. Section 3 discusses interestingness measures for mining association rules. Section 4 presents the lattice and hash tables, an algorithm for fast building the lattice is also discussed in this section. Section 5 presents an algorithm for generating association rules with their measure values using the

Table 6
Results of generating association rules from the lattice in Fig. 2 with lift measure.

Itemset	Sup	**Queue**	Rules with lift measure
D	4	DW,CD,CDW	$\mathrm{D} \xrightarrow{3,9/10} \mathrm{W}, \mathrm{D} \xrightarrow{4,1} \mathrm{C}, \mathrm{D} \xrightarrow{3,9/10} \mathrm{CW}$
DW	3	CDW	$\mathrm{DW} \xrightarrow{3,1} \mathrm{C}$
CDW	3		
CD	4	CDW	$\mathrm{CD} \xrightarrow{3,9/10} \mathrm{W}$
T	4	AT, TW, CT, ATW, ACT, CTW, ACTW	$\mathrm{T} \xrightarrow{3,9/8} \mathrm{A}, \mathrm{T} \xrightarrow{3,9/10} \mathrm{W}, \mathrm{T} \xrightarrow{4,1} \mathrm{C}, \mathrm{T} \xrightarrow{3,9/8} \mathrm{AW}, \mathrm{T} \xrightarrow{3,9/8} \mathrm{AC}, \mathrm{T} \xrightarrow{3,9/10} \mathrm{CW}, \mathrm{T} \xrightarrow{3,9/8} \mathrm{ACW}$
AT	3	ATW, ACT, ACTW	$\mathrm{AT} \xrightarrow{3,6/5} \mathrm{W}, \mathrm{AT} \xrightarrow{3,1} \mathrm{C}, \mathrm{AT} \xrightarrow{3,6/5} \mathrm{CW}$
ATW	3	ACTW	$\mathrm{ATW} \xrightarrow{3,1} \mathrm{C}$
ACTW	3		
ACT	3	ACTW	$\mathrm{ACT} \xrightarrow{3,6/5} \mathrm{W}$
CTW	3	ACTW	$\mathrm{CTW} \xrightarrow{3,3/2} \mathrm{A}$
TW	3	ATW, CTW, ACTW	$\mathrm{TW} \xrightarrow{3,3/2} \mathrm{A}, \mathrm{TW} \xrightarrow{3,1} \mathrm{C}, \mathrm{TW} \xrightarrow{3,3/2} \mathrm{AC}$
CT	4	ACT,CTW, ACTW	$\mathrm{CT} \xrightarrow{3,9/8} \mathrm{A}, \mathrm{CT} \xrightarrow{3,9/10} \mathrm{W}, \mathrm{CT} \xrightarrow{3,9/8} \mathrm{AW}$
A	4	AT, AW, AC, ATW, ACT, ACW, ACTW	$\mathrm{A} \xrightarrow{3,9/8} \mathrm{T}, \mathrm{A} \xrightarrow{4,3/2} \mathrm{W}, \mathrm{A} \xrightarrow{4,1} \mathrm{C}, \mathrm{A} \xrightarrow{3,3/2} \mathrm{TW}, \mathrm{A} \xrightarrow{3,3/2} \mathrm{CT}, \mathrm{A} \xrightarrow{4,6/5} \mathrm{CW}, \mathrm{A} \xrightarrow{3,3/2} \mathrm{CTW}$
AW	4	ATW, ACW, ACTW	$\mathrm{AW} \xrightarrow{3,9/8} \mathrm{T}, \mathrm{AW} \xrightarrow{4,1} \mathrm{C}, \mathrm{AW} \xrightarrow{3,9/8} \mathrm{CT}$
ACW	4	ACTW	$\mathrm{ACW} \xrightarrow{3,9/8} \mathrm{T}$
AC	4	ACT, ACW, ACTW	$\mathrm{AC} \xrightarrow{4,9/8} \mathrm{T}, \mathrm{AC} \xrightarrow{4,6/5} \mathrm{W}, \mathrm{AC} \xrightarrow{3,3/2} \mathrm{TW}$
W	5	DW, TW, AW, CW, CDW, ATW, CTW, ACW, ACTW	$\mathrm{W} \xrightarrow{3,9/10} \mathrm{D}, \mathrm{W} \xrightarrow{3,9/10} \mathrm{T}, \mathrm{W} \xrightarrow{4,6/5} \mathrm{A}, \mathrm{W} \xrightarrow{5,1} \mathrm{C}, \mathrm{W} \xrightarrow{3,9/10} \mathrm{CD}, \mathrm{W} \xrightarrow{3,6/5} \mathrm{AT}, \mathrm{W} \xrightarrow{3,9/10} \mathrm{CT}, \mathrm{W} \xrightarrow{4,6/5} \mathrm{AC}, \mathrm{W} \xrightarrow{3,6/5} \mathrm{ACT}$
CW	5	CDW, CTW, ACW, ACTW	$\mathrm{CW} \xrightarrow{3,9/10} \mathrm{D}, \mathrm{CW} \xrightarrow{3,9/10} \mathrm{T}, \mathrm{CW} \xrightarrow{4,6/5} \mathrm{A}, \mathrm{CW} \xrightarrow{3,6/5} \mathrm{AT}$
C	6	CD, CT, AC, CW, CDW, ACT, CTW, ACW, ACTW	$\mathrm{C} \xrightarrow{4,1} \mathrm{D}, \mathrm{C} \xrightarrow{4,1} \mathrm{T}, \mathrm{C} \xrightarrow{5,1} \mathrm{W}, \mathrm{C} \xrightarrow{3,1} \mathrm{DW}, \mathrm{C} \xrightarrow{3,1} \mathrm{AT}, \mathrm{C} \xrightarrow{3,1} \mathrm{TW}, \mathrm{C} \xrightarrow{3,1} \mathrm{TW}, \mathrm{C} \xrightarrow{4,1} \mathrm{AW}, \mathrm{C} \xrightarrow{3,1} \mathrm{ATW}$

lattice and hash tables. Section 6 presents experimental results, and we conclude our work in section 7.

2. Related work

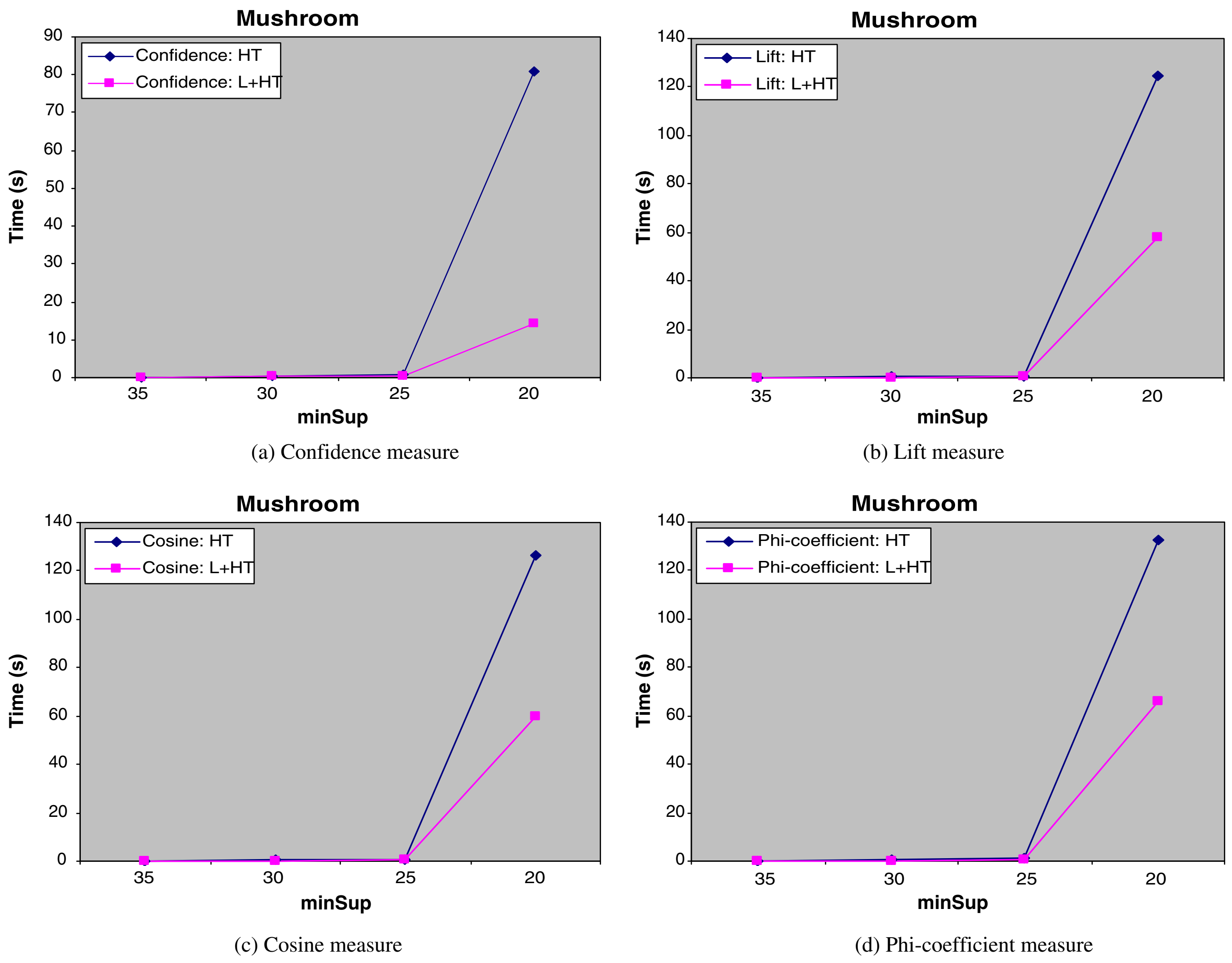
There are many studies in interestingness measures. In 1991, Piatetsky–Shapiro proposed the statistical independence of rules which is the interestingness measure (Piatetsky-Shapiro, 1991). After that, many measures were proposed. In 1994, Agrawal and Srikant proposed the support and the confidence measures for mining association rules (Agrawal & Srikant, 1994). Apriori algorithm for mining rules was discussed. Lift and $\chi 2$ as correlation measures were proposed (Brin et al., 1997). Hilderman and Hamilton, Tan et al. compared differences of interestingness measures and addressed the concept of null-transactions (Hilderman & Hamilton, 2001;Tan et al., 2002). Lee et al. and Omiecinski addressed that all-confidence, coherence, and cosine are null-invariant (Lee et al., 2003; Omiecinski, 2003), and they are good measures for mining correlation rules in transaction databases. Tan et al. discussed the properties of twenty-one interestingness measures and analyzed the impacts of candidates pruning based on the support threshold (Tan et al., 2002). Shekar and Natarajan proposed three measures for getting the relations between item pairs (Shekar & Natarajan, 2004). Besides, giving a lot of measures, some researches have proposed how to choose the measures for a given database (Aljandal et al., 2008; Lenca et al., 2008; Tan et al., 2002).

In building lattice, there are a lot of studies. However, in frequent (closed) itemsets lattice (FIL/FCIL), to our best knowledge, there are three researches: (i) Zaki and Hsiao proposed CHARM-L, an extended of CHARM to build frequent closed itemsets lattice (Zaki & Hsiao, 2005); (ii) Vo and Le proposed the algorithm for building frequent itemsets lattice and based on FIL, they proposed the algorithm for fast mining traditional association rules (Vo & Le, 2009); (iii) Vo and Le proposed an extension of the work in Vo and Le (2009) for building a modification of FIL, they also proposed an algorithm for mining minimal non-redundant association rules (pruning rules generated from the confidence measure) (Vo & Le, 2011).

3. Association rules and interestingness measures

3.1. Association rules mining

Association rule is an expression form $X \overset{q,vm}{\longrightarrow} Y (X \cap Y = \emptyset)$, where q = support (XY) and vm is a measure value. For example, in tradi-



(a) Confidence measure (b) Lift measure

(c) Cosine measure (d) Phi-coefficient measure

Fig. 4. Comparing of the mining time between HT and L + HT in Mushroom database.

tional association rules, *vm* is confidence of the rule and *vm* = support (XY)/support (X).

To fast mine traditional association rules (mining rule with the confidence measure), we can use hash tables (Han & Kamber, 2006). Vo and Le presented a new method for mining association rules using FIL (Vo & Le, 2009). The process includes two phases: (i) Building FIL; (ii) Generating association rules from FIL. This method is faster than that of using hash tables in all of experiments. However, using lattice is hard for determining the support (Y) (the right hand side of the rule), therefore, we need use both lattice and hash tables to determine the supports of X, Y, and XY. With X and XY, we use lattice as in Vo and Le (2009) and use hash tables to determine the support of Y.

3.2. Interestingness measures

We can formula the measure value as follow: Let $vm(n, n_X, n_Y, n_{XY})$ be the measure value of rule $X \to Y$, *vm* value can be computed when we know the measure that needs be computed based on (n, n_X, n_Y, n_{XY}).

Example 1. Consider the example database With $X = \mathrm{AC}$, $Y = \mathrm{TW} \Rightarrow n = 6,\ n_X = 4,\ n_Y = 3,\ n_{XY} = 3 \Rightarrow n_{\overline{X}} = 2,\ n_{\overline{Y}} = 3$.

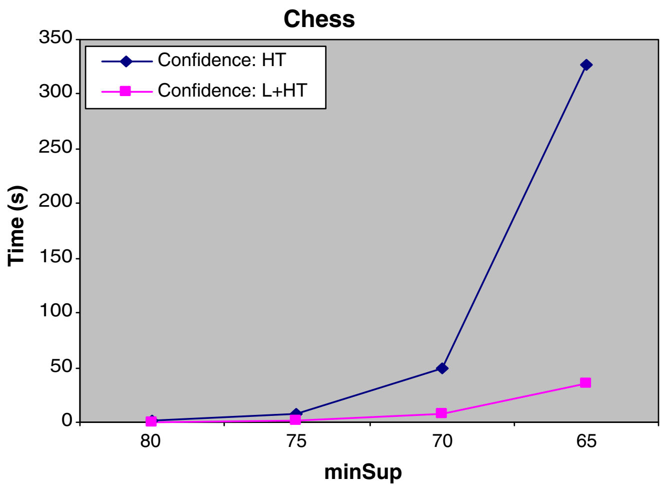
We have the values of some measures in Table 2.

4. Lattice and hash tables

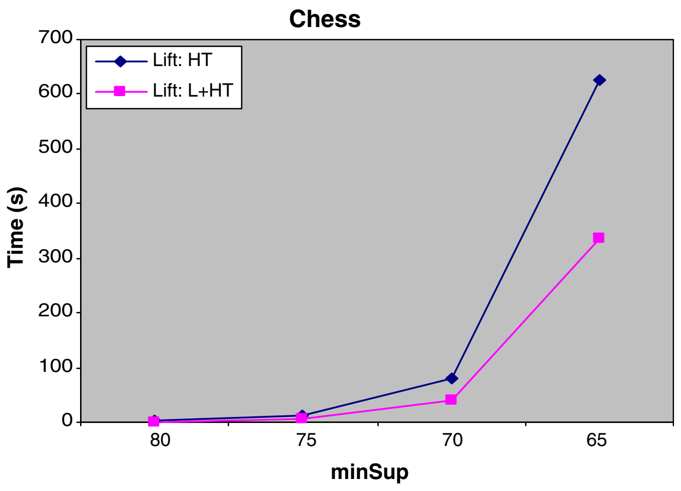
4.1. Building FIL

Vo and Le presented an algorithm for fast building FIL, we present it here to make reader easier to read next sections (Vo & Le, 2009).

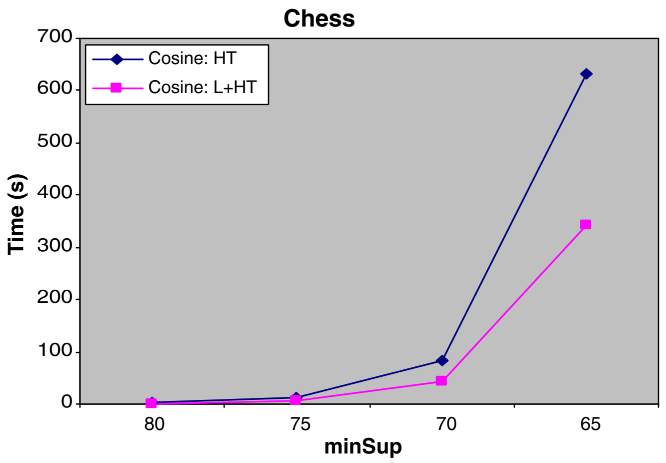
At first, the algorithm initializes the equivalence class $[\emptyset]$ which contains all frequent 1-itemsets. Next, it calls **ENUMERATE_LATTICE**($[\boldsymbol{P}]$) function to create a new frequent itemset by combining two frequent itemsets of equivalence class $[\boldsymbol{P}]$, and produces a lattice node $\{\boldsymbol{I}\}$ (if $\boldsymbol{I}$ is frequent). The algorithm will add a new node $\{\boldsymbol{I}\}$ into a set of child nodes of both $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$, because $\{\boldsymbol{I}\}$ is a direct child node of both $\boldsymbol{l}_i$ and $\boldsymbol{l}_j$. Especially, the rest child nodes of $\{\boldsymbol{I}\}$ must be the child nodes of child node $\boldsymbol{l}_i$, so **UPDATE_LATTICE** function only considers $\{\boldsymbol{I}\}$ with ***lcc*** nodes that are also child nodes of the node $\boldsymbol{l}_i$, if $\boldsymbol{lcc} \supset \boldsymbol{I}$ then $\{\boldsymbol{I}\}$ is parent node of $\{\boldsymbol{lcc}\}$. Finally, the result will be the root node $\boldsymbol{l}_r$ of the lattice. In fact, in case of mining all itemsets from the database, we can assign the minSup equal to 1 (see Fig. 1).



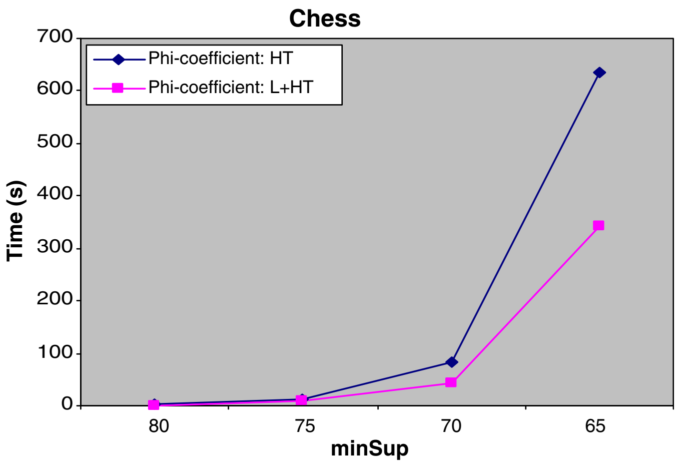
(a) Confidence measure



(b) Lift measure



(c) Cosine measure



(d) Phi-coefficient measure

Fig. 5. Comparing of the mining time between using HT and using L + HT in Chess database.

4.2. An example

Fig. 2 illustrates the process of building frequent itemsets lattice from the database in Table 1. First, the root node of lattice ($\boldsymbol{L}_r$) contains frequent 1-itemset nodes. Assume that we have lattice nodes {D}, {T}, {DW}, {CD}, {CDW}, {AT}, {TW}, {CT}, {ATW}, {ACT}, and {ACTW} (which contains in dash polygon). Consider the process of producing lattice node {AW}: Because of $\boldsymbol{l}_i = \{A\}$ and $\boldsymbol{l}_j = \{W\}$, the algorithm only considers {AW} with the child nodes of {AT} ({A} only has one child node {AT} now):

- Consider {ATW}: since AW $\subset$ ATW, {ATW} is a child node of {AW}.
- Consider {ACT}: since AW $\not\subset$ ACT, {ACT} is not a child node of {AW}.

The dark-dash links represent the path that points to child nodes of {AW}. The dark links represent the process of producing {AW} and linking {AW} with its child nodes. The lattice nodes enclosed in the dash polygon represents lattice nodes that considered before producing node {AW}.

4.3. Hash tables

To mine association rules, we need determine the support of X, Y and XY. With X and XY, we can use the FIL as mentioned above. The support of Y can be determined by using hash tables. We use two levels of hash tables: (i) The first level: using the length of itemset as a key; (ii) In case of the itemsets with the same length, we use hash tables with key which is computed by $\sum_{y \in Y} y$ (Y is the itemset which need determine the support).

Example 2. Consider the database given in Table 1 with minSup = 50%, we have all frequent itemsets as follows:Table 3 contains frequent itemsets from the database in Table 1 with minSup = 50% and Table 4 illustrates the keys of itemsets in Table 3. In fact, based on Apriori property, the length of itemsets increases from 1 to k (where k is the longest itemset). Therefore, we need not use hash table in level 1. By the length, we can use a suitable hash table. Besides, to avoid the case of different itemsets which have the same key, we use prime numbers to be the keys of single items as in Table 5.

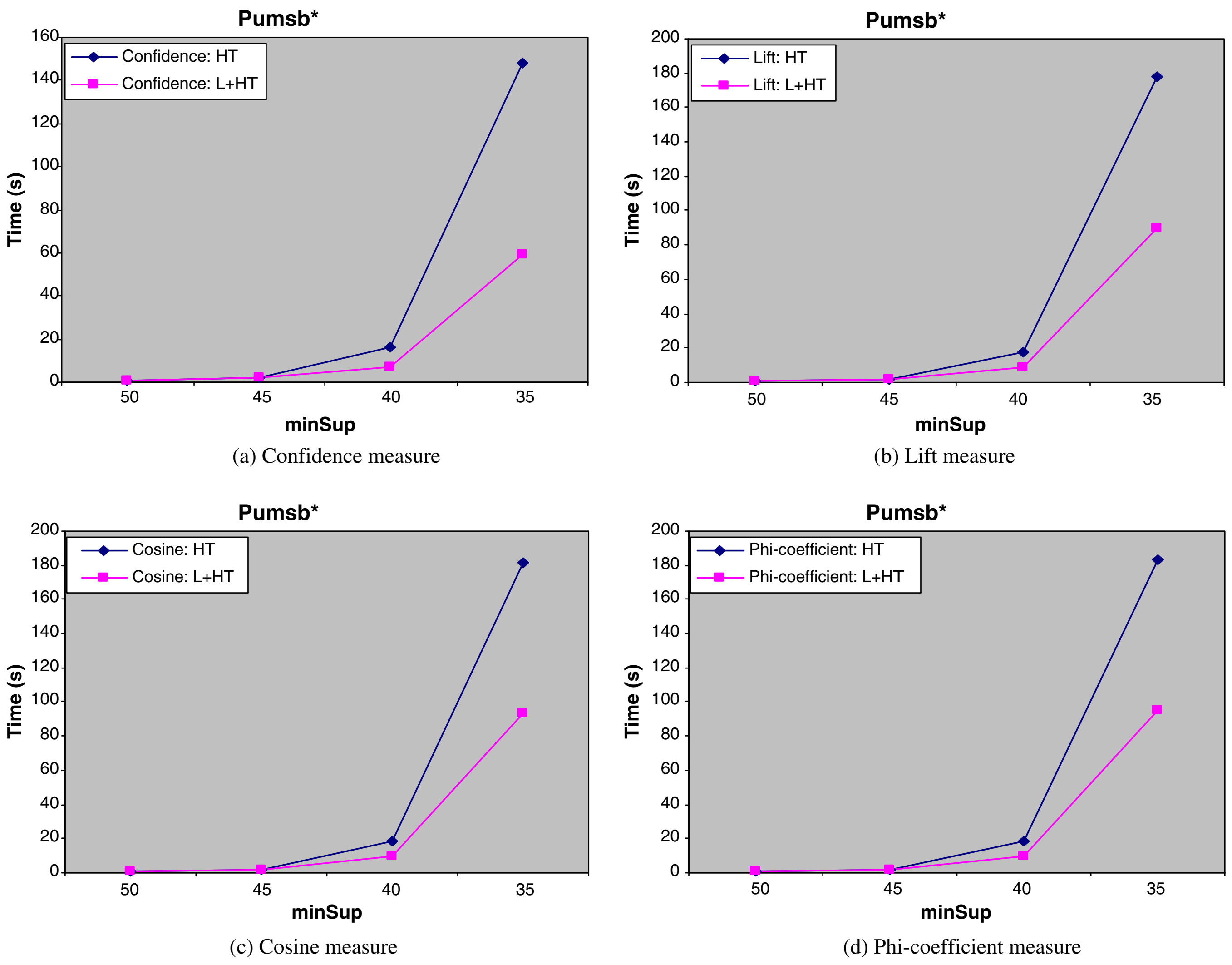


Fig. 6. Comparing of the mining time between using HT and using L + HT in Pumsb* database.

We can see that keys of itemsets in the same hash table are not equal as in Table 5. Therefore, the time for getting the support of itemset is often O (1).

5. Mining association rules with interestingness measures

This section presents an algorithm for mining association rules with a given interestingness measure. First of all, we traverse the lattice to determine X, XY and their supports. With Y, we compute $k = \sum_{y \in Y} y$ (y is a prime number or an integer number). Based on its length and its key, we can get the support.

5.1. Algorithm for mining association rules and their interestingness measures

Fig. 3 presents an algorithm for mining association rules with interestingness measures using lattice and hash tables. At first, the algorithm traverses all child nodes $\boldsymbol{L_c}$ of the root node $\boldsymbol{L_r}$, and then it calls **EXTEND_AR_LATTICE**($\boldsymbol{L_c}$) function to traverse all nodes in the lattice (recursively and mark in the visited nodes if flag turns on). Considering **ENUMERATE_AR**($\boldsymbol{L_c}$) function, it uses a queue for traversing all child nodes of $\boldsymbol{L_c}$ (and marking all visited nodes for rejecting coincides). For each child node (of $\boldsymbol{L_c}$), we compute the measure value by using $vm(n,n_X,n_Y,n_{XY})$ function (where n is the number of transactions, n_X = support ($\boldsymbol{L_c}$), n_{XY} = support ($\boldsymbol{L}$) and n_Y = get support from the hash table $|Y|^{th}$ with $Y = \boldsymbol{L} \backslash \boldsymbol{L_c}$), and add this rule into **ARs**. In fact, the number of generated rules is very large. Therefore, we need use a threshold to reduce the rules set.

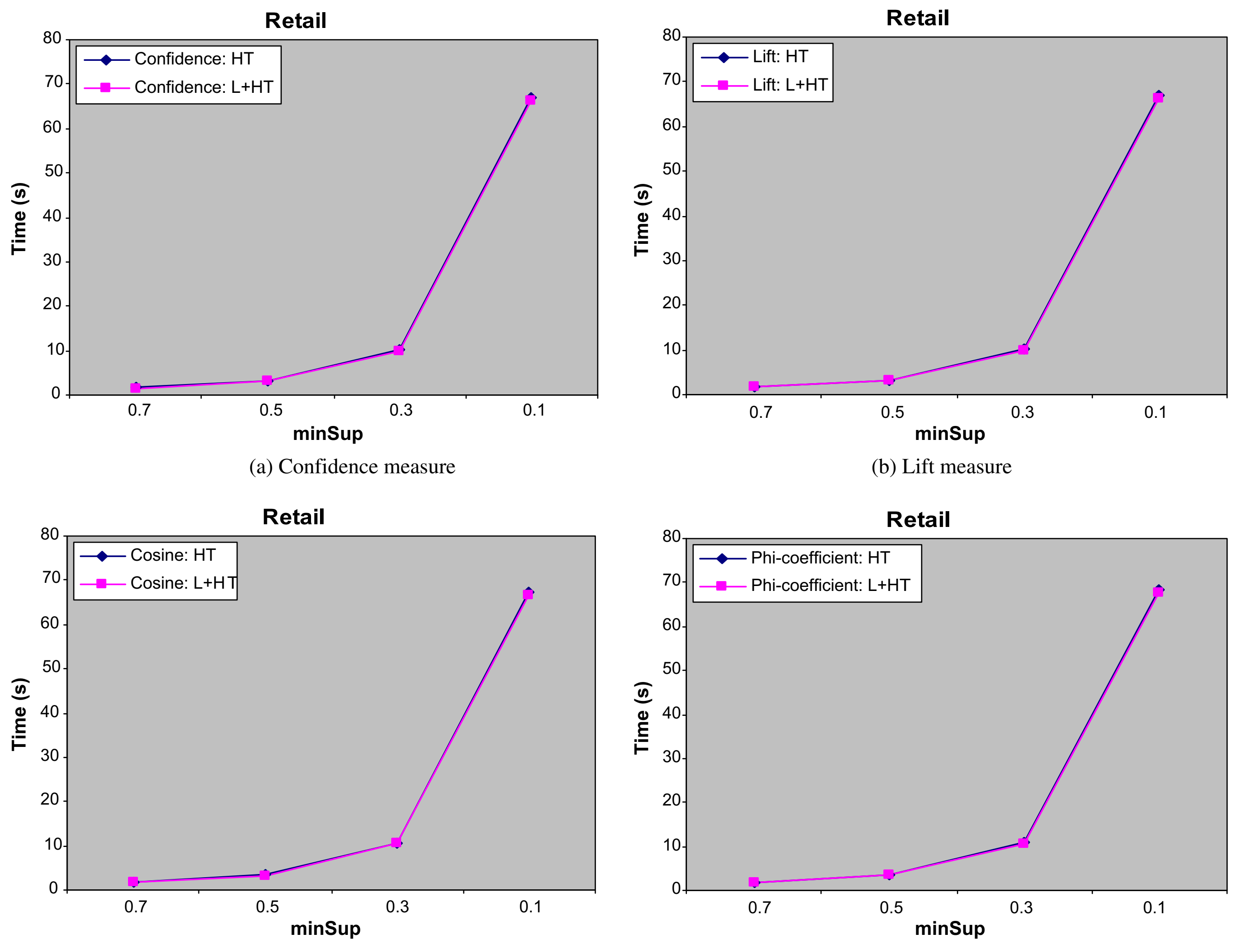
5.2. An example

Table 6 shows the results of generating association rules from the lattice in Fig. 2 with lift measure. We have 60 rules corresponding to lift measure. If minLift = 1.1, we have 30 rules that satisfy minLift. Consider the process of generating association rules from node $\boldsymbol{L_c}$ = D of the lattice (Fig. 2), we have (n_X = support (D) = 4):

At first, **Queue** = $\emptyset$. The child nodes of D are {DW,CD}, they are added into **Queue** $\Rightarrow$ **Queue** = {DW, CD}.

Because **Queue** $\neq \emptyset \Rightarrow \boldsymbol{L}$ = DW (**Queue** = {CD}):

- n_{XY} = support (L) = 3
- Because $Y = \boldsymbol{L} - \boldsymbol{L_c}$ = W $\Rightarrow n_Y$ = (Get the support from HashTables[1] with key = 11) = 5 $\Rightarrow vm(6,4,5,3) = \frac{6*3}{4*5} = \frac{9}{10}$ (using lift measure).



(a) Confidence measure (b) Lift measure

(c) Cosine measure (d) Phi-coefficient measure

Fig. 7. Comparing of the mining time between using HT and using L + HT in retail database.

- Add all child nodes of CD (only CDW) into **Queue** and mark node CDW $\Rightarrow$ **Queue** = {CD, CDW}.
  Next, because **Queue** $\neq \emptyset \Rightarrow$ ***L*** = CD (**Queue** = {CDW}):
- n_{XY} = support (L) = 4
- Because $Y = \boldsymbol{L} - \boldsymbol{L}_c = C \Rightarrow n_Y$ = (Get the support from HashTables[1] with key = 3) = 6 $\Rightarrow vm(6,4,6,4) = \frac{6*4}{4*6} = 1$. Next, because **Queue** $\neq \emptyset \Rightarrow$ ***L*** = CDW (**Queue** = $\emptyset$):
- n_{XY} = support (L) = 3
- Because $Y = \boldsymbol{L} - \boldsymbol{L}_c = CW \Rightarrow n_Y$ = (Get support from HashTables[2] with key = 14) = 5 $\Rightarrow vm(6,4,5,3) = \frac{6*3}{4*5} = \frac{9}{10}$.

  Next, because **Queue** = $\emptyset$, stop.

6. Experimental results

All experiments described below have been performed on a centrino core 2 duo (2 × 2.53 GHz) with 4 GBs RAM, running Windows 7, and algorithms were coded in C# (2008). The experimental databases were downloaded from http://fimi.cs.helsinki.fi/data/ to use for experiments, their features are shown in Table 7.

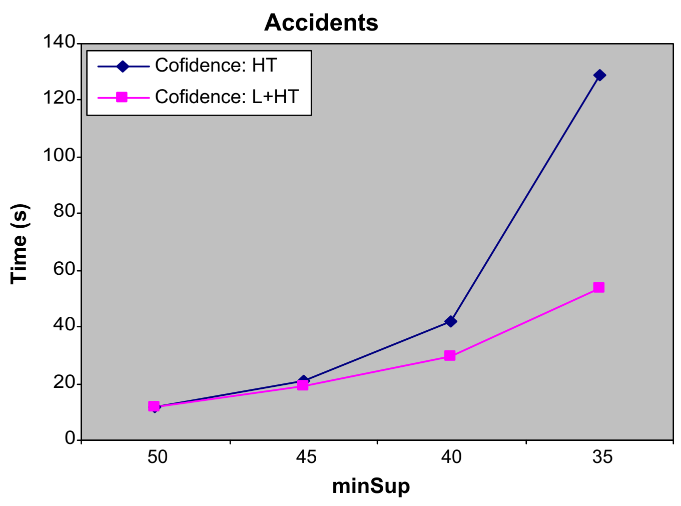
We test the proposed algorithm in many databases. Mushroom and Chess have few items and transactions in that Chess is dense database (more items with high frequent). The number of items in Accidents database is medium, but the number of transactions is large. Retail has more items, and its number of transactions is medium.

Numbers of rules generated from databases are very large. For example: consider database Pumsb* with minSup = 35%, the number of frequent itemsets is 116747 and the number of association rules is 49886970 (Table 8).

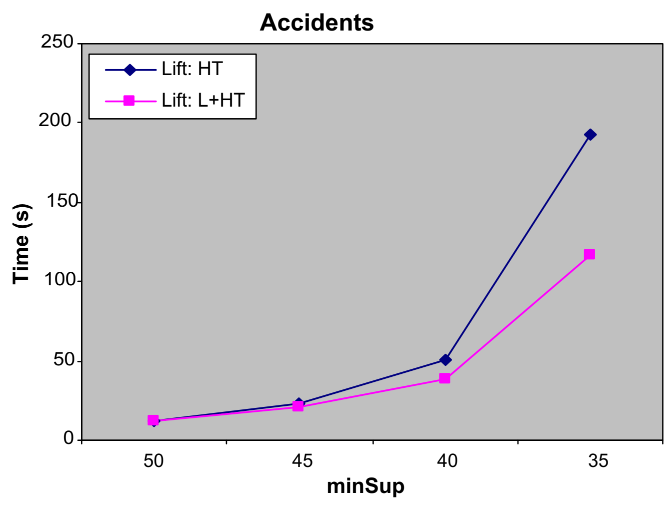
6.1. The mining time using hash tables and using both lattice and hash tables

Figures from 4 to 8 compare the mining time between using HT (hash tables) and using L + HT (combination between lattice and hash tables).

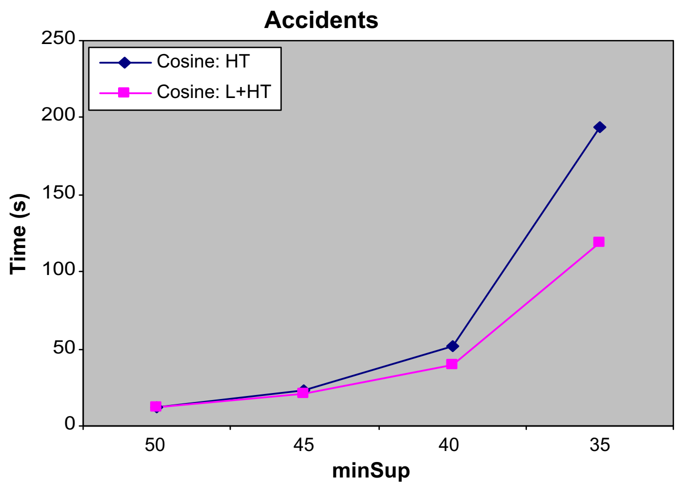
Results in Fig. 4(a) compare the mining time between HT and L + HT in confidence measure. Figs. 4 (b,c,d) are for lift, cosine and phi-coefficient measures corresponding. Experimental results from Fig. 4 show that the mining time of combination between L + HT is always faster than that of using only HT. For example: with minSup = 20% in Mushroom, if we use confidence measure, the mining time of using L + HT is 14.13 and of using HT is 80.83,



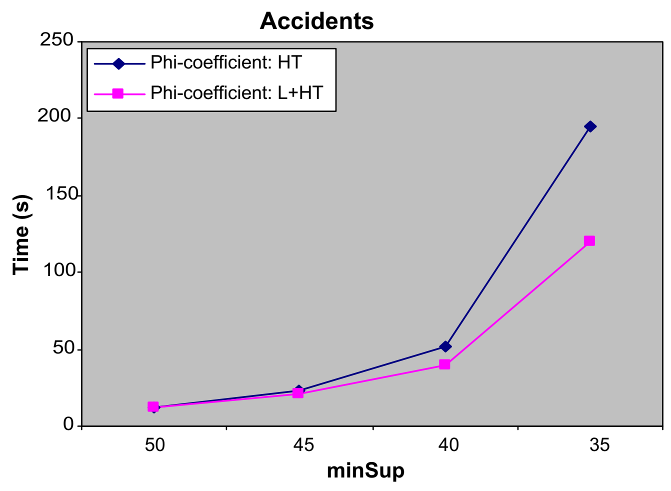
(a) Confidence measure



(b) Lift measure

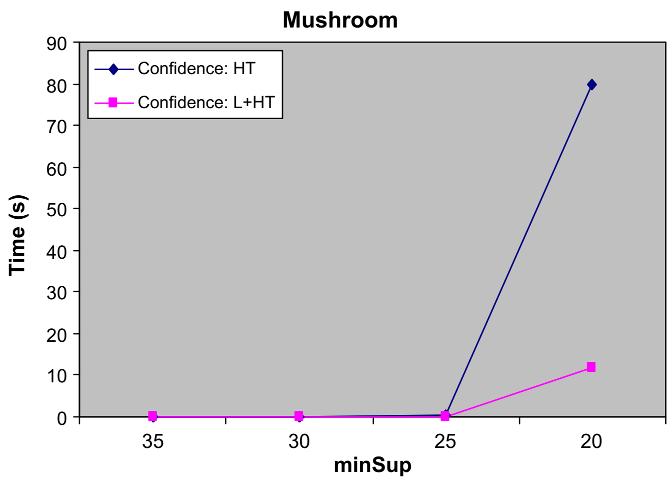


(c) Cosine measure

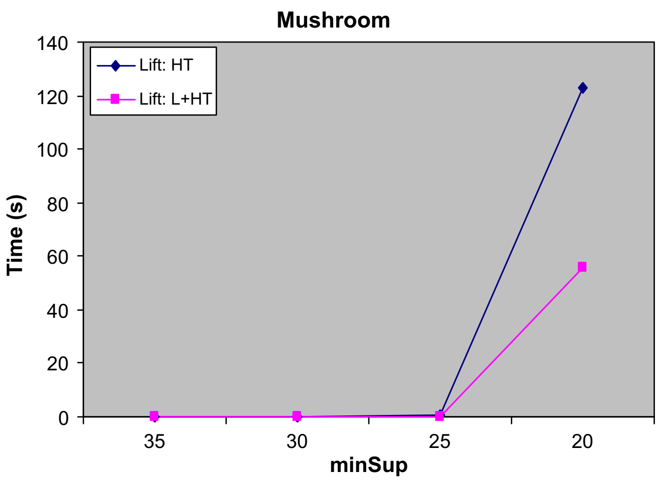


(d) Phi-coefficient measure

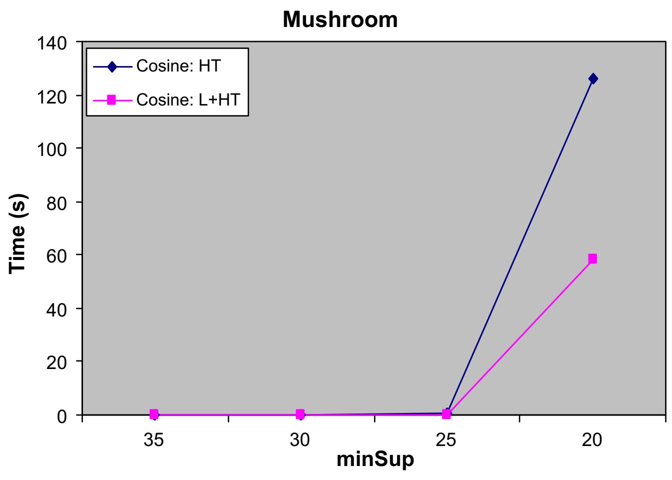
Fig. 8. Comparing of the mining time between using HT and using L + HT in accidents database.



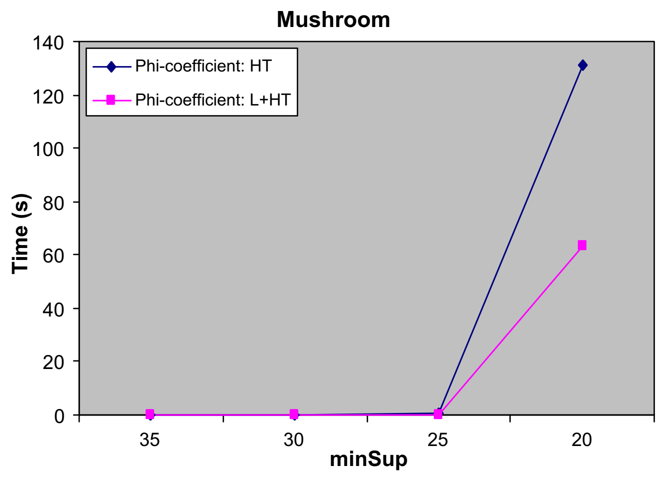
(a) Confidence measure



(b) Lift measure



(c) Cosine measure



(d) Phi-coefficient measure

Fig. 9. Comparing of the mining time between using HT and using L + HT in Mushroom database (without computing the time of mining frequent itemsets and buiding lattice).

the scale is $\frac{14.13}{80.83} \times 100\% = 17.48\%$. If we use lift measure, the scale is $\frac{57.81}{124.43} \times 100\% = 46.31\%$, the scale of cosine measure is $\frac{59.91}{126.57} \times 100\% = 47.33\%$ and of phi-coefficient is $\frac{65.79}{132.49} \times 100\% = 49.66\%$. The scale of the confidence measure is the smallest because it need not use HT to determine the support of Y (the right hand side of rules).

Experimental results from Figs. 4–8 show that the mining time using L + HT is always faster than that of using only HT. The more decreasing minSup is, the more efficient of the mining time that uses L + HT is (Retail has a little change when we decrease the minSup because it contains a few rules).

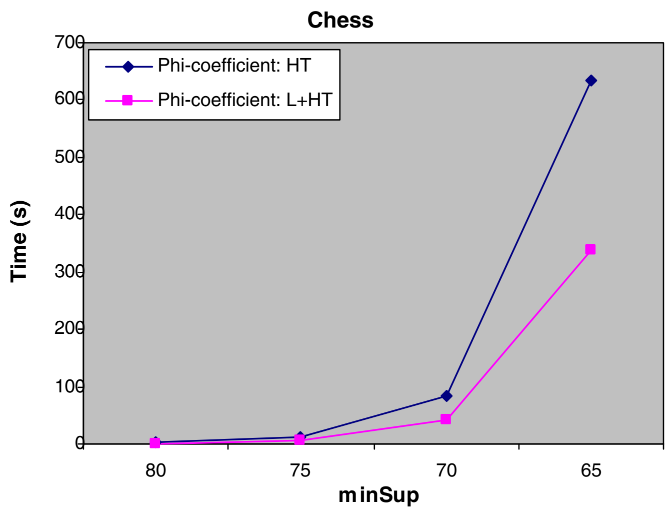
6.2. Without computing the time of mining frequent itemsets and building lattice

The mining time in section 6.1 is the total time of mining frequent itemsets and generating rules (using HT) and that of building lattice and generating rules (using L + HT). If we ignore the time of mining frequent itemsets and buiding lattice, we have results as in Figs. 9 and 10.

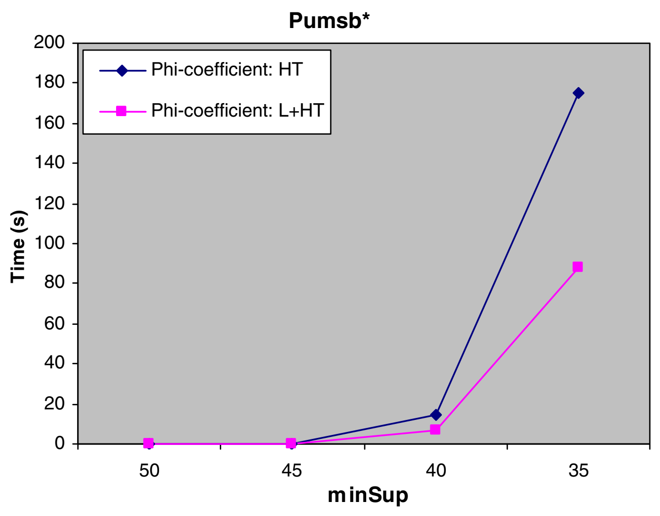
From Fig. 9, with minSup = 20%, if we use the confidence measure, the mining time of combination between L + HT is 11.989 and the mining time using HT is 79.69, the scale is $\frac{11.989}{79.69} \times 100\% = 15.05\%$ (compare to 17.48 of Fig. 4(a), it is more efficient). If we use lift measure, the scale is $\frac{55.439}{123.14} \times 100\% = 45.02\%$, the scale of cosine measure is $\frac{58.139}{125.84} \times 100\% = 46.20\%$ and of phi-coefficient is $\frac{63.339}{131.04} \times 100\% = 48.34\%$. Results in Fig. 9 show that the scale between using L + HT and using only HT decreases in case of ignoring the time of mining frequent itemsets and buiding lattice. Therefore, if we mine frequent itemsets or buiding lattice one time, and use results for generating rules many times, then using L + HT are more efficient.

7. Conclusion and future work

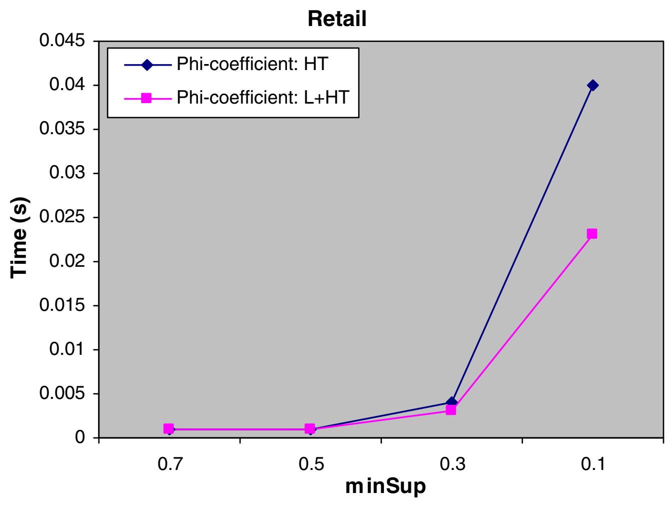
In this paper, we proposed a new method for mining association rules with interestingness measures. This method uses lattice and hash tables to compute the interestingness measure values fast. Experimental results show that the proposed method is very efficient when compares with only using hash tables. With itemset X and itemset XY, we get their supports by traversing the lattice and mark all traversed nodes. With itemset Y, we use hash tables to get its support. When we only compare the time of generating rules, the scale in using lattice and hash tables is more efficient



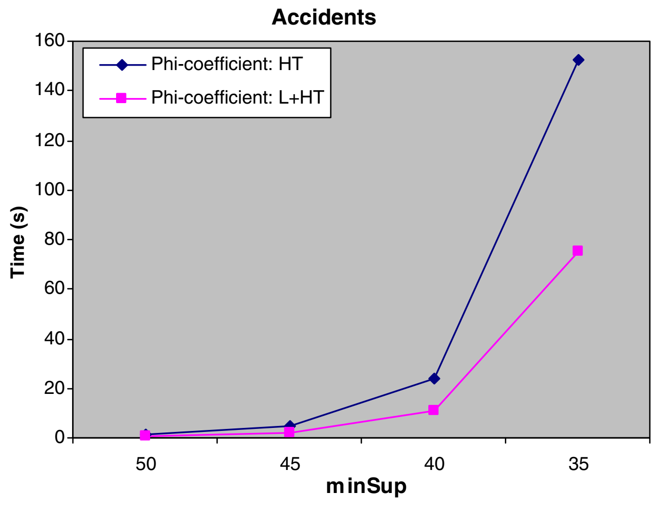
(a) Chess database



(b) Pumsb* database



(c) Retail database



(d) Accidents database

Fig. 10. Comparing of the mining time between HT and L + HT with phi-coefficient measure (without computing the time of mining frequent itemsets and buiding lattice).

than that of using only hash tables. Besides, we can use the gotten itemsets to compute the values of many different measures. Therefore, we can use this method for integrating interestingness measures. In the future, we will study and propose an efficient algorithm for selecting *k* best interestingness rules based on lattice and hash tables.

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. In *VLDB'94* (pp. 487–499).

Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD conference Washington, DC, USA, May 1993* (pp. 207–216).

Aljandal, W., Hsu, W. H., Bahirwani, V., Caragea, D., & Weninger, T. (2008). Validation-based normalization and selection of interestingness measures for association rules. In *Proceedings of the 18th international conference on artificial neural networks in engineering (ANNIE 2008)* (pp. 1–8).

Athreya, K. B., & Lahiri, S. N. (2006). *Measure theory and probability theory*. Springer-Verlag.

Bayardo, R. J., Agrawal, R. (1999). Mining the most interesting rules. In *Proceedings of the fifth ACM SIGKDD* (pp. 145–154).

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket analysis. In *Proceedings of the 1997 ACM-SIGMOD international conference on management of data (SIGMOD'97)* (pp. 255–264).

Freitas, A. A. (1999). On rule interestingness measures. *Knowledge-based Systems, 12*(5–6), 309–315.

Grahne, G., & Zhu, J. (2005). Fast algorithms for frequent itemset mining using FP-trees. *IEEE Transactions on Knowledge and Data Engineering, 17*(10), 1347–1362.

Han, J., & Kamber, M. (2006). *Data mining: Concept and techniques* (2nd ed.). Morgan Kaufman Publishers. pp. 239–241.

Hilderman, R., & Hamilton, H. (2001). *Knowledge discovery and measures of interest*. Kluwer Academic.

Holena, M. (2009). Measures of ruleset quality for general rules extraction methods. *International Journal of Approximate Reasoning (Elsevier), 50*(6), 867–879.

Huebner, R. A. (2009). Diversity-based interestingness measures for association rule mining. In *Proceedings of ASBBS* (Vol. 16, p. 1). Las Vegas.

Huynh, H. X., Guillet, F., Blanchard, J., Kuntz, P., Gras, R., & Briand, H. (2007). *A graph-based clustering approach to evaluate interestingness measures: A tool and a comparative study. Quality measures in data mining*. Springer-Verlag. pp. 25–50.

Lee, Y. K., Kim, W. Y., Cai, Y., & Han, J. (2003). CoMine: Efficient mining of correlated patterns. In *Proceeding of ICDM'03* (pp. 581–584).

Lenca, P., Meyer, P., Vaillant, P., & Lallich, S. (2008). On selecting interestingness measures for association rules: User oriented description and multiple criteria decision aid. *European Journal of Operational Research, 184*(2), 610–626.

MCGarry, K. (2005). *A survey of interestingness measures for knowledge discovery. Knowledge engineering review*. Cambridge University Press. pp. 1–24.

Omiecinski, E. (2003). Alternative interest measures for mining associations. *IEEE Transactions on Knowledge and Data Engineering, 15*, 57–69.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. *Knowledge Discovery in Databases*, 229–248.

Shekar, B., & Natarajan, R. (2004). A transaction-based neighborhood-driven approach to quantifying interestingness of association rules. In *Proceedings of ICDM'04*.

Steinbach, M., Tan, P. N., Xiong, H., & Kumar, V. (2007). *Objective measures for association pattern analysis*. American Mathematical Society.

Tan, P. N., Kumar, V., & Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In *Proceeding of the ACM SIGKDD international conference on knowledge discovery in databases (KDD'02)* (pp. 32–41).

Vo, B., & Le, B. (2009). Mining traditional association rules using frequent itemsets lattice. In *39th international conference on CIE, July 6–8, Troyes, France* (pp. 1401–1406).

Vo, B., & Le, B. (2011). Mining minimal non-redundant association rules using frequent itemsets lattice. *Journal of Intelligent Systems Technology and Applications, 10*(1), 92–106.

Waleed, A. A. (2009). *Itemset size-sensitive interestingness measures for association rule mining and link prediction* (pp. 8–19). Ph.D dissertation, Kansas State University.

Wang, J., Han, J., & Pei, J. (2003). CLOSET+: Searching for the best strategies for mining frequent closed itemsets. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 236–245).

Yafi, E., Alam, M. A., & Biswas, R. (2007). Development of subjective measures of interestingness: From unexpectedness to shocking. *World Academy of Science, Engineering and Technology, 35*, 88–90.

Yao, Y., Chen, Y., & Yang, X. (2006). A measurement-theoretic foundation of rule interesting evaluation. *Studies in Computational Intelligence (Book Chapter), 9*, 41–59.

Zaki, M. J., & Hsiao, C. J. (2005). Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Transactions on Knowledge and Data Engineering, 17*(4), 462–478.